AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1.   (Currently Amended) A method for moving data objects (201.x) in a computer system (101) from a first (107) storage location to a second (108) storage location, comprising:

a) selecting one or more data objects (201.x) having an identifier (ID) from the first storage location (107) (301)[[,]];

b) storing said the ID in a second lock object 204) (302)[[,]];

c) in case step b) has been performed successfully: determining whether the ID is stored successfully in the second lock object and, upon a successful storage, storing said the ID in a first lock object (203) (307)[[,]];

d) storing a data object (201.x), the ID of which is contained in the first lock object (203), at the second storage location (108) (405)[[,]];

e) deleting [[a]] the data object (201.x), the ID of which is contained in the first lock object (203), from said the first storage location (107) (502)[[,]];

f) deleting [[an]] the ID from the first lock object (203) earliest at a time at which step e) for after the respective data object (201.x) assigned to that ID has been completed (503)[[,]] deleted from the first storage location; and

g) deleting [[an]] the ID from the second lock object (204) earliest at a time at which step c) for after a particular ID has been completed stored in the first lock object (308).

2.   (Currently Amended) The method of claim 1, wherein ~~a~~ each data object comprises one ~~ore~~ or more fields of one or more tables, ~~(201, 202)~~ and wherein the ID comprises one or more key fields of the one or more tables ~~(201, 202)~~.

3.   (Currently Amended) The method of claim 1 ~~or 2~~, wherein ~~in step d)~~ the data ~~objects (201.x) are~~ object is stored in one or more files ~~(405)~~ and wherein an assignment of the ID to the file or to a name of the file, in which the data object assigned to ~~said~~ the ID is stored, is stored in the first lock object ~~(203) (408)~~.

4.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 3~~, wherein the first lock object ~~(203)~~ is stored on a nonvolatile storage means ~~(107)~~.

5.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 4~~, wherein ~~in step c)~~ the ID is stored in the second lock object ~~(204)~~ immediately after ~~performing step a)~~ the step of selecting one or more data objects having an identifier from the first storage location for the respective data object ~~(201.x)~~.

6.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 4~~, wherein ~~in step b)~~ the ID of the selected data object ~~(201.x)~~ is stored in the second lock object ~~(204) shortly~~ before ~~the storing process according to step d) for~~ the data object ~~(201.x)~~ assigned to that ID is stored ~~is started~~.

7.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 6~~, wherein storing the ID in the first lock object further comprises:

storing ~~in step c) the~~ IDs of all selected data objects ~~(201.x) are stored~~ in the first lock object ~~(203)~~ before ~~the~~ a first storing process according to step d) is started.

8.   (Currently Amended) The method ~~one~~ of ~~claims~~ claim 1 ~~to 7~~, further comprising:

[[h)]] checking ~~before or while performing any of steps a) to c) for a data object (201.x)~~, whether an ID for the data object ~~(201.x)~~ has been stored in a first lock object ~~(203)~~, and if ~~yes~~ <u>the data object has been stored</u>, skipping at least step d) for that data object ~~(201.x)~~.

9.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 8~~, further comprising:

[[i)]] checking ~~before or while performing any of steps a) to d) for a data object (201.x)~~, whether ~~that~~ <u>the</u> data object ~~(201.x)~~ is contained in the second storage location ~~(108)~~, and if ~~yes~~ <u>the data object is contained in the second storage location</u>, skipping at least step d) for that data object ~~(201.x)~~.

10.   (Currently Amended) The method of claim 9, wherein ~~said~~ <u>the</u> checking ~~step i)~~ is performed by querying [[a]] <u>the</u> first lock object ~~(203)~~.

11.   (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 10~~, further comprising:

[[j)]] ~~in case of a failure in step d)~~ <u>determining whether the data object was stored in the first lock object successfully and, upon unsuccessful storage,</u> checking[[,]] whether the data object ~~(201.x)~~ assigned to the respective ID has been completely stored in the second storage location ~~(108)~~, and ~~in case of no~~ <u>if the respective ID has</u>

not been completely stored, skipping at least steps e) and f) for that data object (201.x)

and deleting the ID from the first lock object (203).

12.   (Currently Amended) The method of one-of-claims claim 1 to 11 for use in

an enterprise resource planning software.

13.   (Currently Amended) A computer system (101) for processing data by

means of or in a software application (111), comprising:

[[-]] memory means (112) for storing program instructions;

[[-]]input means (102, 104, 113) for entering data;

[[-]]storage means (107, 108) for storing data;

[[-]]a processor (105) responsive to the program instructions, wherein the [[-]]

program instructions to carry out a method as of any of claims 1 to 12 comprise

program code means for performing a method for moving data objects in the computer

system from a first storage location to a second storage location, the method

comprising:

selecting one or more data objects having an identifier (ID) from the first storage

location;

storing the ID in a second lock object;

determining whether the ID is stored successfully, and upon a successful

storage, storing the ID in a first lock object;

storing a data object, the ID of which is contained in the first lock object, at the

second storage location;

deleting the data object, the ID of which is contained in the first lock object, from the first storage location;

deleting the ID from the first lock object after the respective data object assigned to that ID has been deleted; and

deleting the ID from the second lock object after a particular ID has been stored in the first lock object.

14. (Cancelled)

15. (Currently Amended) A computer readable medium comprising ~~program code~~ instructions for performing a method ~~as of any of claims 1 to 12 if said program code is executed on a computer system~~ for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting one or more data objects having an identifier (ID) from the first storage location;

storing the ID in a second lock object;

determining whether the ID is stored successfully, and upon a successful storage, storing the ID in a first lock object;

storing a data object, the ID of which is contained in the first lock object, at the second storage location;

deleting the data object, the ID of which is contained in the first lock object, from the first storage location;

deleting the ID from the first lock object after the respective data object assigned to that ID has been deleted; and

deleting the ID from the second lock object after a particular ID has been stored in the first lock object.

16.   (Cancelled)

17.   (New) The computer readable medium of claim 15, wherein each data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

18.   (New) The computer readable medium of claim 15, wherein the data object is stored in one or more files and wherein an assignment of the ID to the file or to a name of the file, in which the data object assigned to the ID is stored, is stored in the first lock object.

19.   (New) The computer readable medium of claim 15, wherein the first lock object is stored on a nonvolatile storage means.

20.   (New) The computer readable medium of claim 15, wherein the ID is stored in the second lock object immediately after selecting one or more data objects having an identifier from the first storage location for the respective data object.

21.   (New) The computer readable medium of claim 15, wherein the ID of the selected data object is stored in the second lock object before the data object assigned to that ID is stored.

22.   (New) The computer readable medium of claim 15, wherein storing the ID in the first lock object further comprises:

storing IDs of all selected data objects in the first lock object before storing the

data object at the second storage location.

23.   (New) The computer readable medium of claim 15, further comprising:

checking whether an ID for the data object has been stored in a first lock object,

and if the data object has been stored, skipping the storing of the data object, the ID of

which is contained in the first lock object, at the second storage location, for that data

object.

24.   (New) The computer readable medium of claim 15, further comprising:

checking whether the data object is contained in the second storage location,

and if the data object is contained in the second storage location, skipping the storing of

the data object, the ID of which is contained in the first lock object, at the second

storage location, for that data object.

25.   (New) The computer readable medium of claim 15, wherein the checking

is performed by querying the first lock object.

26.   (New) The computer readable medium of claim 15, further comprising:

determining whether the data object was stored in the first lock object

successfully, and upon unsuccessful storage, checking whether the data object

assigned to the respective ID has been completely stored in the second storage

location, and if the respective ID has not been completely stored, skipping the deleting

of the data object from the first storage location and the deleting of the ID from the first

lock object after the respective data object assigned to that ID has been deleted for that data object, and deleting the ID from the first lock object.

27.   (New) The computer readable medium of claim 15, wherein the computer readable medium is provided as part of a computer program product.

28.   (New) A computerized system for processing data, comprising:

means for selecting one or more data objects having an identifier (ID) from the first storage location;

means for storing the ID in a second lock object;

means for determining whether the ID is stored successfully, and upon a successful storage, storing the ID in a first lock object;

means for storing a data object, the ID of which is contained in the first lock object, at the second storage location;

means for deleting the data object, the ID of which is contained in the first lock object, from the first storage location;

means for deleting the ID from the first lock object after the respective data object assigned to that ID has been deleted; and

means for deleting the ID from the second lock object after a particular ID has been stored in the first lock object.

29.   (New) The computer system of claim 28, wherein each data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

30.   (New) The computer system of claim 28, wherein the data object is stored in one or more files and wherein an assignment of the ID to the file or to a name of the file, in which the data object assigned to the ID is stored, is stored in the first lock object.

31.   (New) The computer system of claim 28, wherein the first lock object is stored on a nonvolatile storage means.

32.   (New) The computer system of claim 28, wherein the ID is stored in the second lock object immediately after the means for selecting one or more data objects having an identifier from the first storage location for the respective data object.

33.   (New) The computer system of claim 28, wherein the ID of the selected data object is stored in the second lock object before the data object assigned to that ID is stored.

34.   (New) The computer system of claim 28, wherein the means for storing the ID in a first lock object further comprises:

means for storing IDs of all selected data objects in the first lock object before storing the data object at the second storage location.

35.   (New) The computer system of claim 28, wherein the means for storing the ID in a first lock object further comprises:

means for checking whether an ID for the data object has been stored in a first lock object, and if the data object has been stored, skipping the storing of the data object, the ID of which is contained in the first lock object, at the second storage location, for that data object.

36.   (New) The computer system of claim 28, further comprising:

means for checking whether the data object is contained in the second storage

location, and if the data object is contained in the second storage location, skipping the

storing of the data object, the ID of which is contained in the first lock object, at the

second storage location, for that data object.

37.   (New) The computer system of claim 28, wherein the means for checking

comprises means for querying the first lock object.

38.   (New) The computer system of claim 28, further comprising:

means for determining whether the data object was stored in the first lock object

successfully;

means for checking, upon unsuccessful storage in the first lock object, whether

the data object assigned to the respective ID has been completely stored in the second

storage location; and

means for skipping, if the respective ID has not been completely stored, the

deleting of the data object from the first storage location and the deleting of the ID from

the first lock object after the respective data object assigned to that ID has been deleted

for that data object, and deleting the ID from the first lock object.